Introduction to Machine Learning and Deep Learning

Part 1

Till Sauerwein and Konrad U. Förstner

ZB MED – Information Centre for Life Science & TH Köln

Workshop *Omics data analytics and AI approaches (Machine Learning and Deep learning)*,
GATC conference 2024, New Delhi

2024-04-08 - 2024-04-11

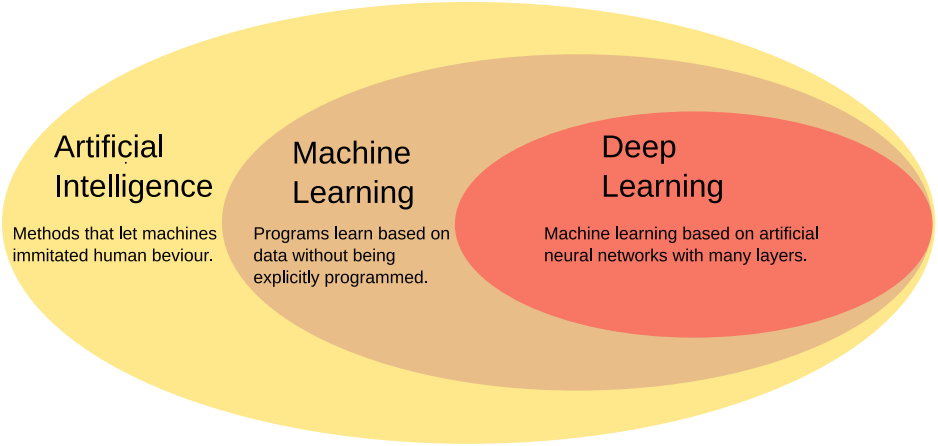
ZB MED Information Centre for Life Sciences
Technology Arts Sciences TH Köln

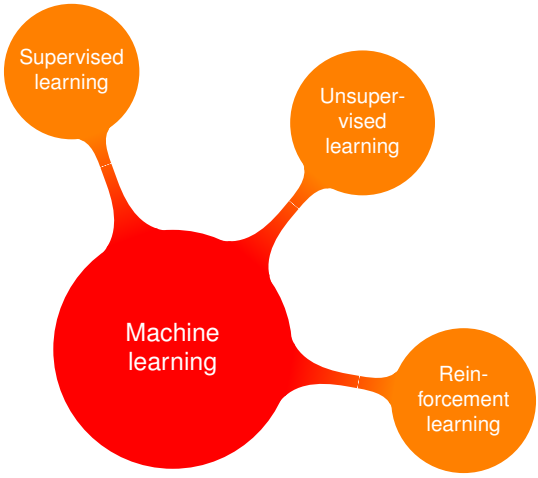
After the lecture you should have a basic understanding of machine learning / deep learning approaches and potential applications in biological research.

After the practical part you should be able to implement them with Python and the packages scikit-learn and keras/tensorflow.

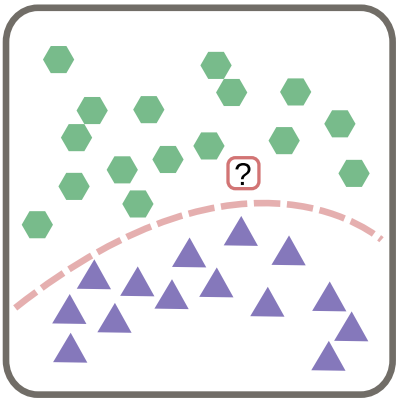
We will not cover the mathematical background in depth. This is not needed at this level but recommended later.

AI encompasses many methods

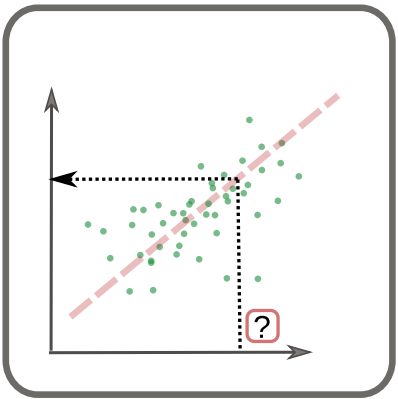



Supervised learning
Unsuper-vised learning
Machine learning
Rein-forcement learning

Two types of tasks that can be solved with supervised learning

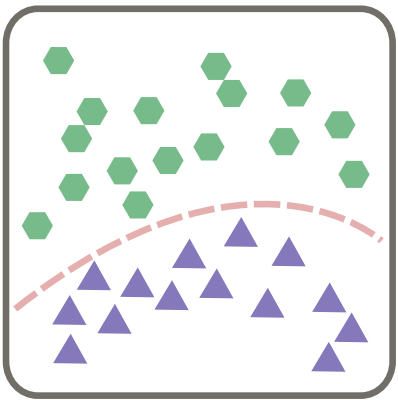


Classification

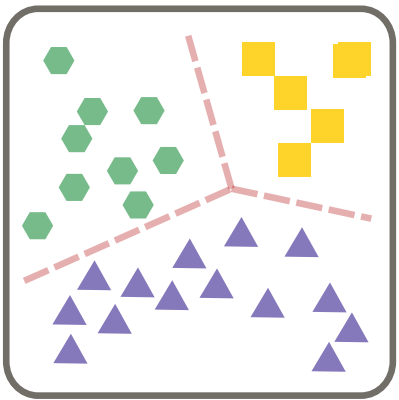


Regression

Classification types



Binary classification



Multi-class classification

Supervised learning: Generate models that generalize from given examples.

Basic concept of supervised machine learning

The model / function maps from a given two-dimensional matrix X to an output vector y with labels (classification) or numerical values (regression).

$$X_1 \rightarrow y_1$$
$$X_2 \rightarrow y_2$$
$$X_3 \rightarrow y_3$$

In the actual training / learning process the parameters of the model / function are estimated. The model is then able to project the input variable X to the output variable y.

$$y = f(X)$$

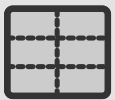
Example of classification

Cancer classification based on single-cell gene expression data.

Example of regression

Predicting the gene expression level of a gene based on the gene expression levels of several regulators.

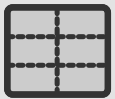
Entities and their features



Entities (aka. samples, data points) are described by **features** (aka. covariates, attributes) that have **values**.

E.g. for different cell lines (entities) the relative expression (values) of several genes (features).

Entities and their features



Features can be

- categorical
  - Nominal (e.g. cell line, cancer type, eye color, gender)
  - Ordinal (e.g. very bad, bad, good, very good)
- numerical
  - Discrete (e.g. gene length in nucleotides, number of cells)
  - Continuous (e.g. cell length, concentration, relative expression)

Feature selection

Choosing features with high variance.

Feature A	Feature B	Feature C	Feature D	...
10.00	5.01	102.01	120	...
20.91	5.01	102.00	200	...
80.03	5.01	102.09	980	...
90.19	5.00	103.00	700	...
50.99	5.02	102.31	703	...
80.63	5.01	102.30	443	...

Feature scaling

Normalizing the feature values to their ranges e.g. min/max normalization, mean normalisation, standard score / z-score normalization.

Feature A	Feature B
4.3	537
5.3	703
2.2	510
1.5	200
5.2	760

$\Rightarrow$

Scaled Feature A	Scaled Feature B
0.736	0.601
1.000	0.898
0.184	0.554
0.000	0.00
0.974	1.00

Features encoding

Translating categorical values into numerical values
(e.g. via one-hot encoding)

	A	**C**	**G**	**T**
A	1	0	0	0
C	0	1	0	0
G	0	0	1	0
T	0	0	0	1

e.g. AATTGC becomes:

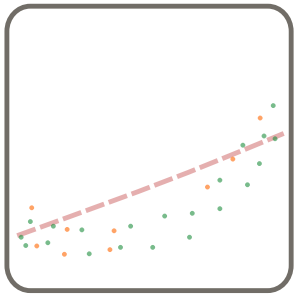
1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0

How well does the model fit?

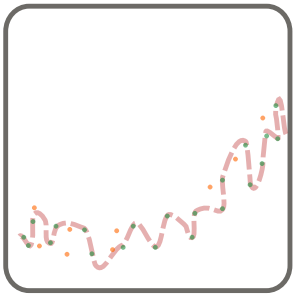
Overfitting: Good performance on the training data, poor generalization to other data

Underfitting: Poor performance on the training data and poor generalization to other data

Regularization: Different methods to prevent overfitting



Underfitting



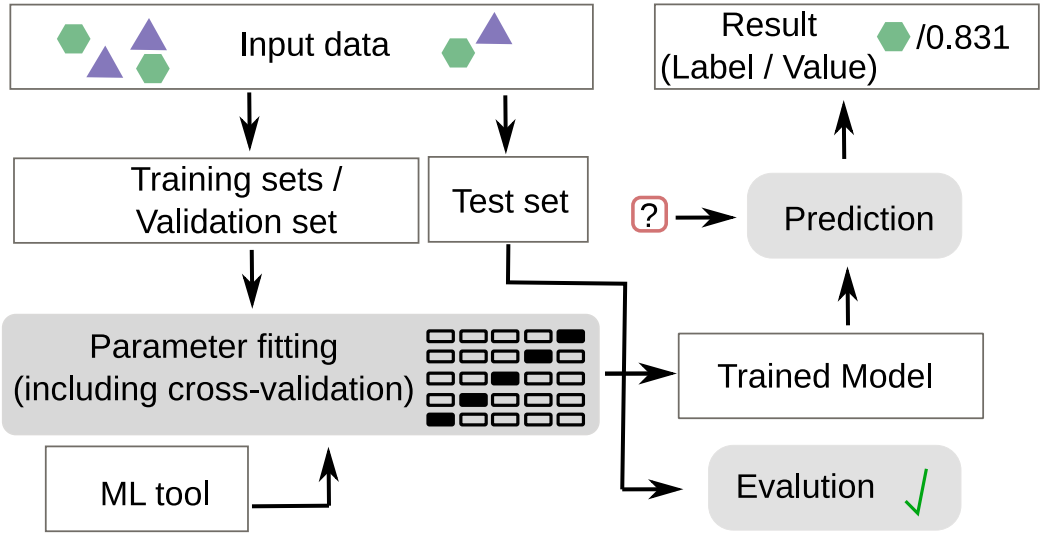
Overfitting

Good fit

Workflow for parameter fitting and evaluation

1.) Split into training and test/validation set (e.g. 75%/25%)
2.) Train model by estimating the parameters with the training set
3.) Evaluate the performance by using the test/validation set (e.g. scored as accuracy)

Workflow with cross-validation



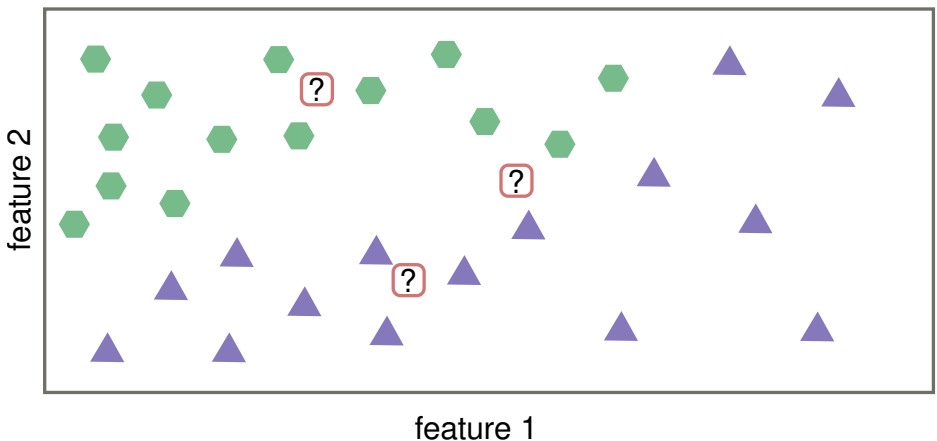
Overview of different methods

- K-Nearest neighbor
- Naive Bayes
- Linear Regression
- Logistic Regression
- Decision trees
- Artificial Neural Network (multilayer perceptron)
- Genetic Programming

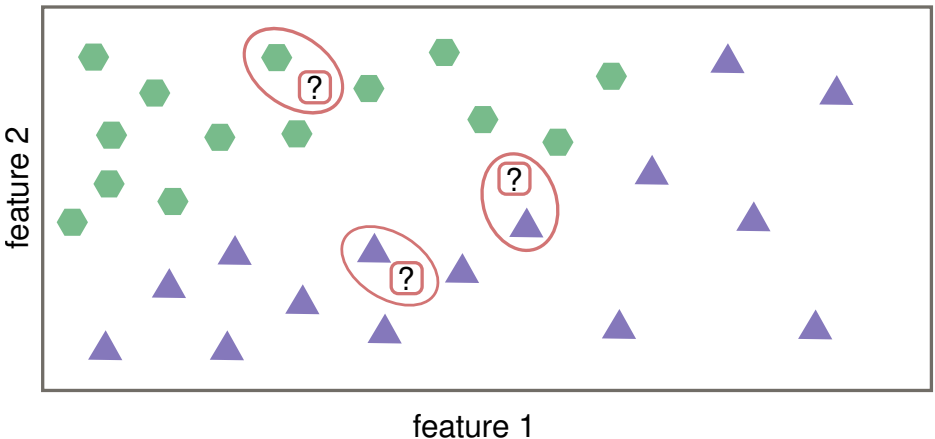
k-Nearest Neighbors

- For classification and regression
- Simplest case of supervised machine learning
- Can be easily applied to multi-class classification

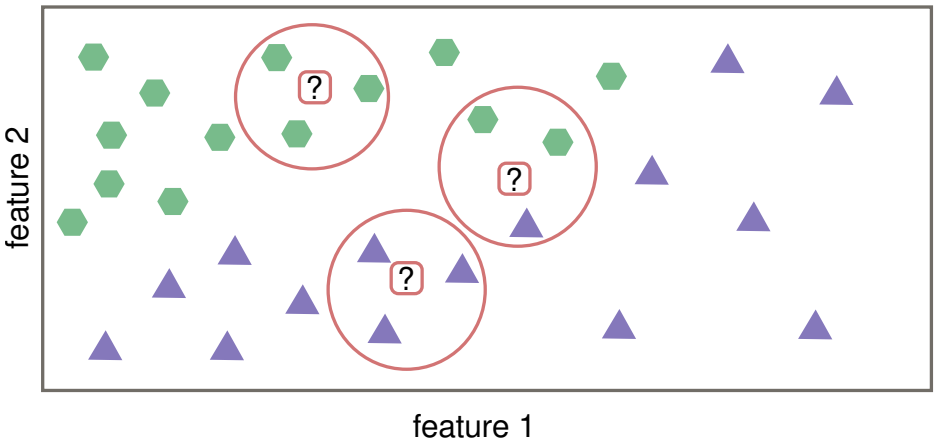
k-Nearest Neighbors



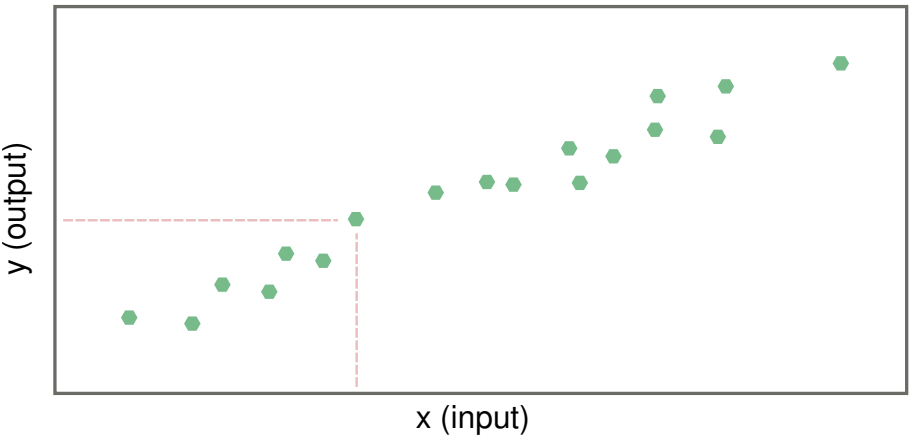
k-Nearest Neighbors



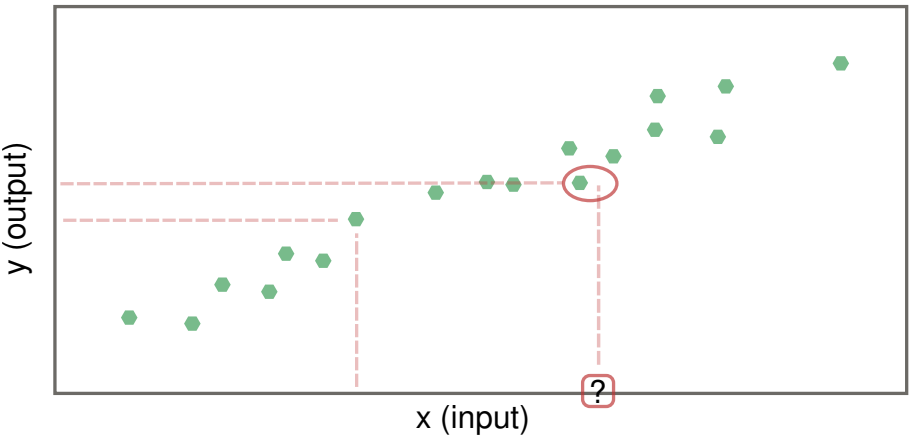
k-Nearest Neighbors



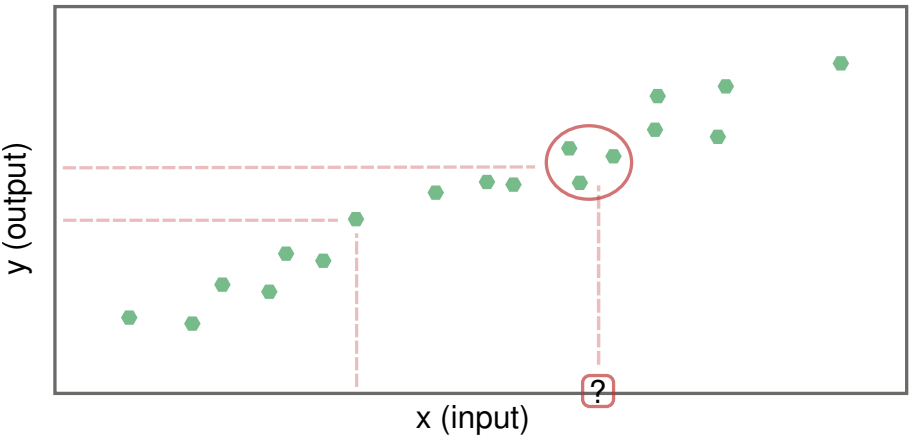
k-Nearest Neighbors



k-Nearest Neighbors



k-Nearest Neighbors



Linear models

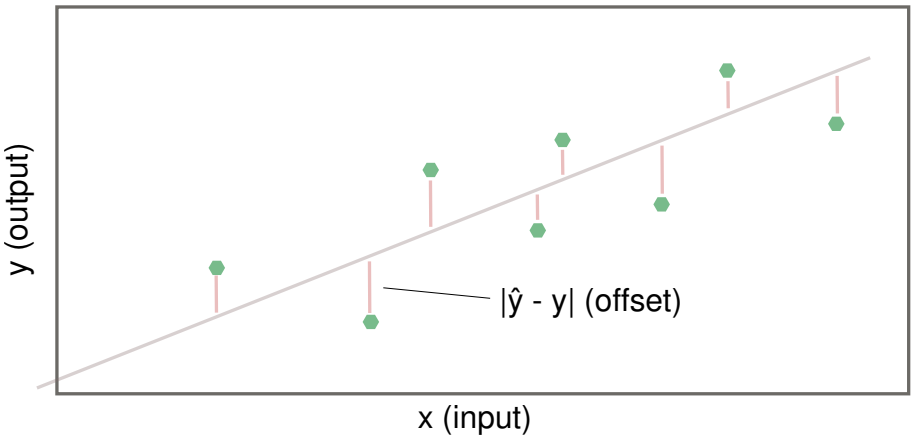
$$\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 + ... + w_n x_n + b$$

with n as the number of features
w are the different weights/coefficients
b the intercept

Different ways to estimate the parameters

- Ordinary Least Squares
  - no parameters - easy to use but no possibility to adapt
- Ridge
  - coefficients should be close to zero
  - more resistant against overfitting
- Least Absolute Shrinkage and Selection Operator (LASSO)

Ordinary least squares (OLS)



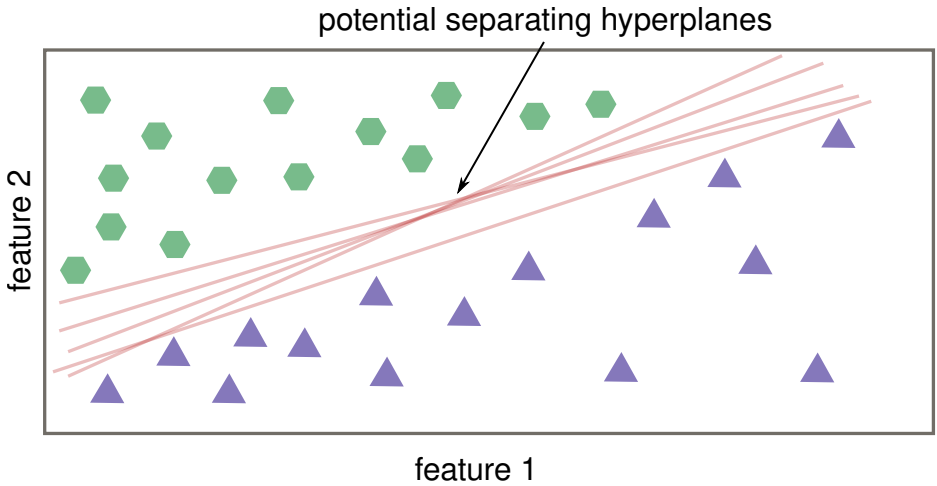
Minimize the offset between $\hat{y}$ and y the
mean squared error (MSE) or sum of squared errors (SSE).

Once the parameters (b and the weights w) of

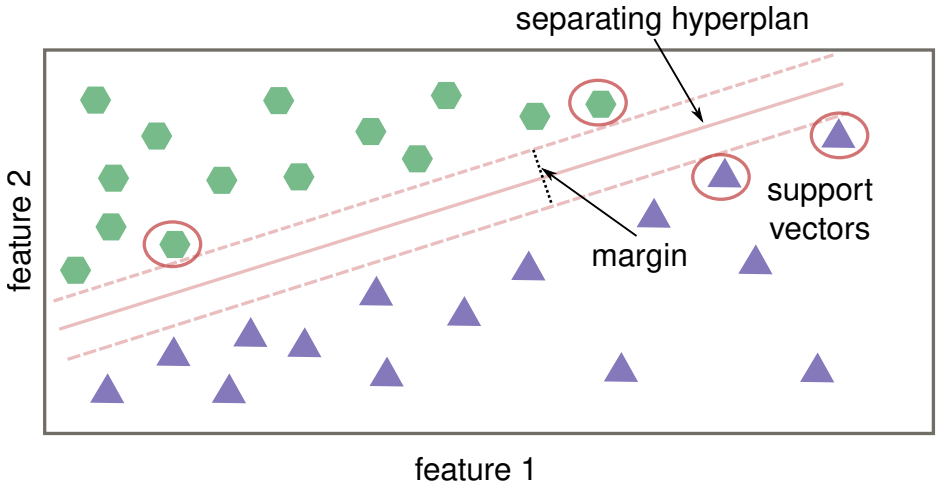
$$\hat{y} = w_1 x_1 + w_2 x_2 + w_3 x_3 + ... + w_n x_n + b$$

are estimated the prediction can be performed by putting the x values of the data points into the equation to predict the y value.

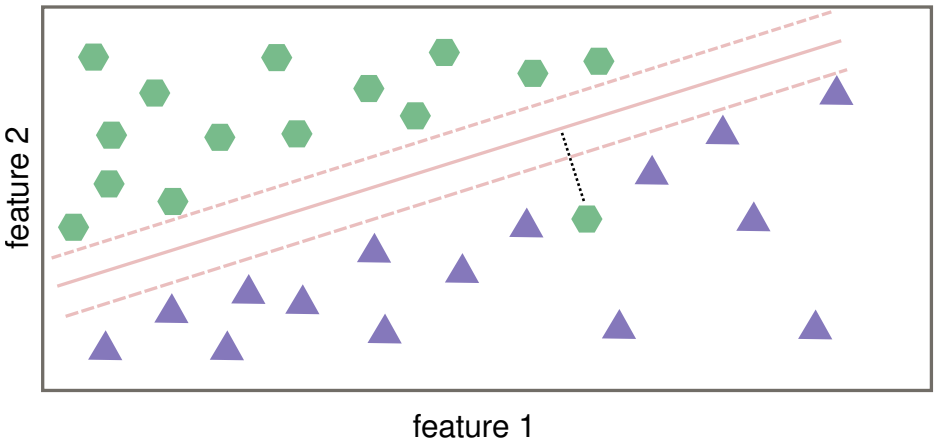
Support Vector Machines (SVMs) – Separating hyperplane



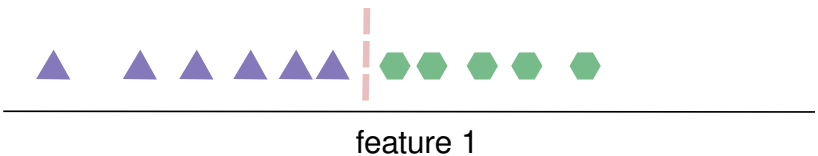
Support Vector Machines (SVMs) – Margin



Support Vector Machines (SVMs) – Soft Margin

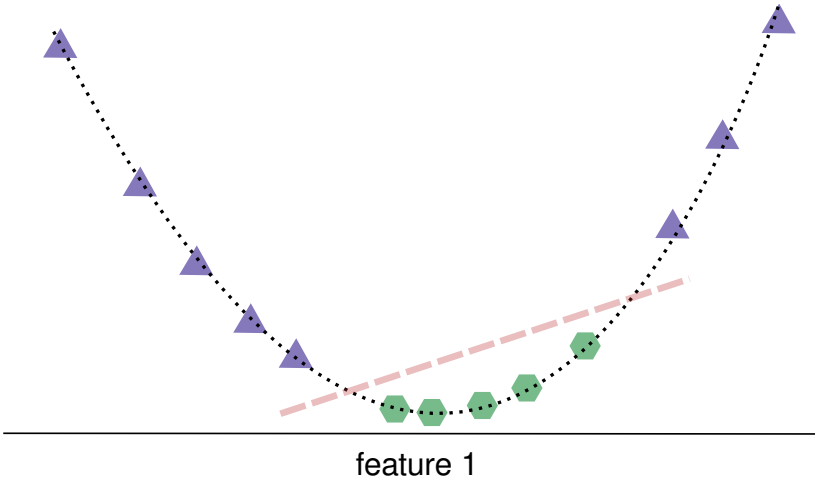


Support Vector Machines (SVMs) – Kernel trick

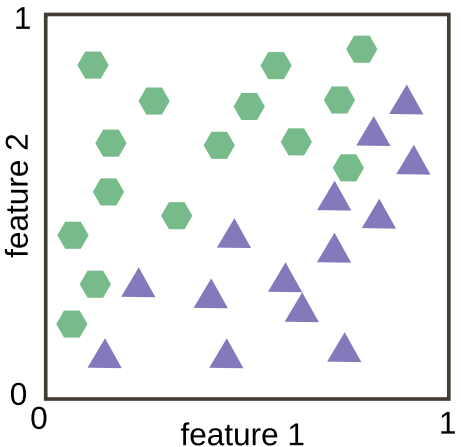


SVM – Kernel trick

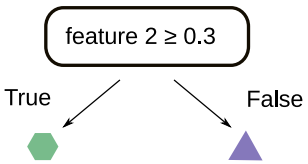
Support Vector Machines (SVMs) – Kernel trick

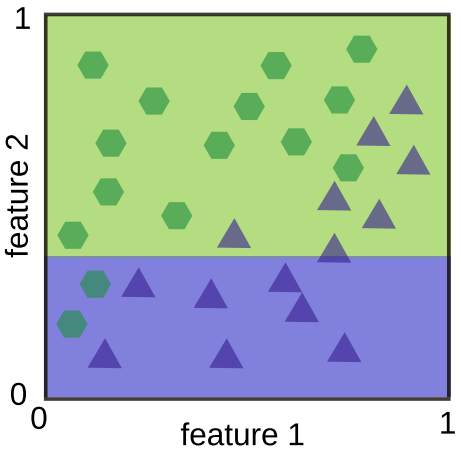


Decision Trees

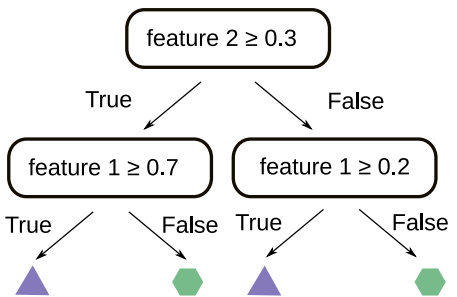


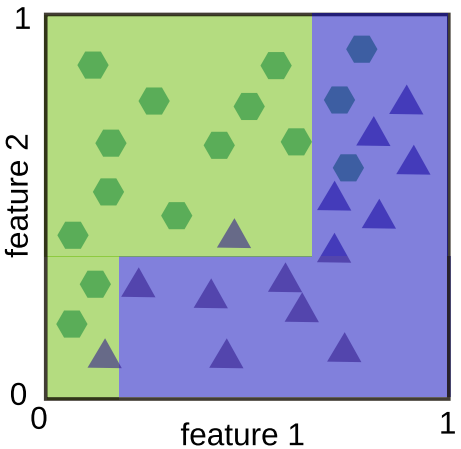
Decision Trees





Decision Trees





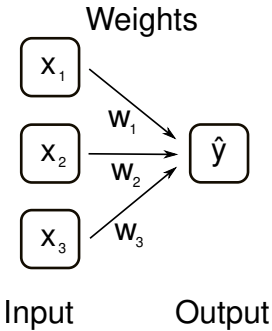
Random forest

- In the random forests approach many different decision trees are generated by a randomized tree-building algorithm.
- The training set is sampled with replacement to produce a modified training set of equal size to the original but with some training items included more than once.
- In addition, when choosing the question at each node, only a small, random subset of the features is considered.
- Decision is happening by presenting the data to all tree and then do a voting.

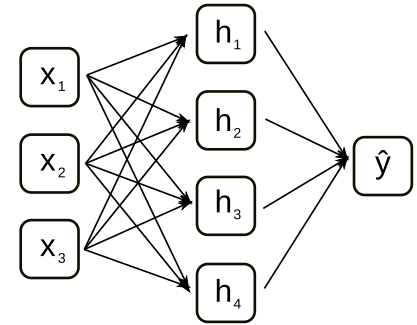
Artificial Neural Networks

- Inspired by natural neural networks
- For classification or regression

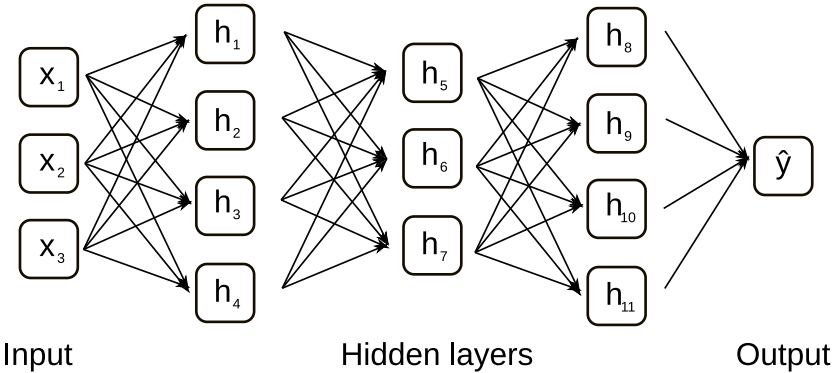
Artificial Neural Networks



Artificial Neural Networks



Artificial Neural Networks



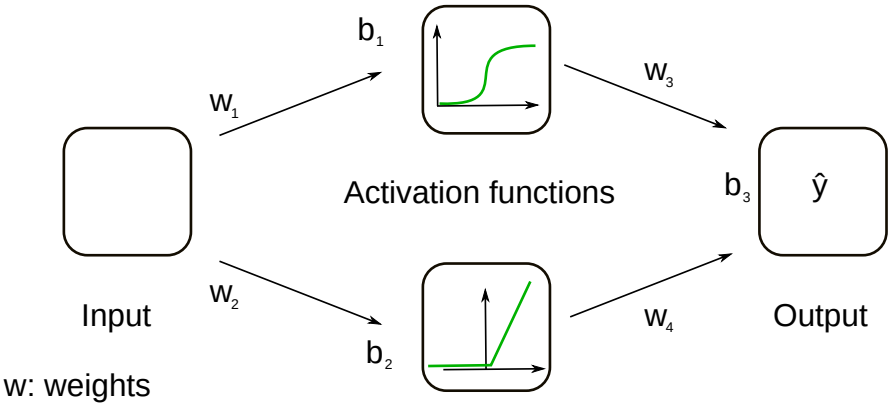
Artificial Neural Networks

For each neuron in an ANN:

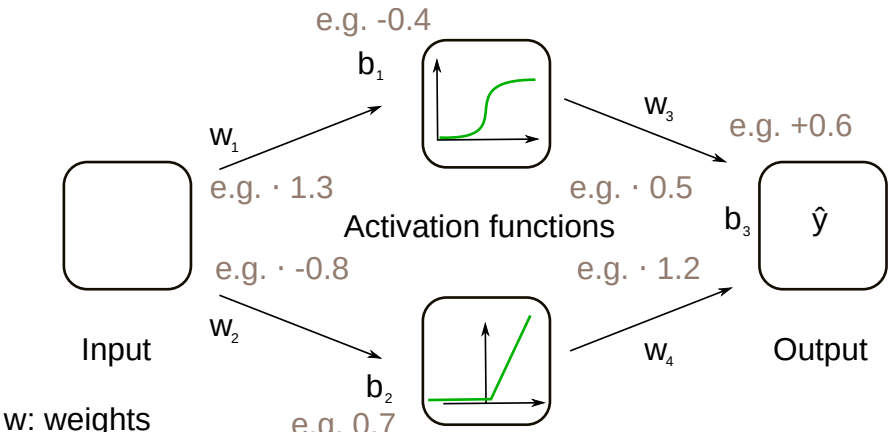
$$y = \sigma\left(\sum_{i=1}^{n} (w_i x_i) + b\right)$$

y = output value
σ = activation function
w = weight
b = bias

Artificial Neural Networks



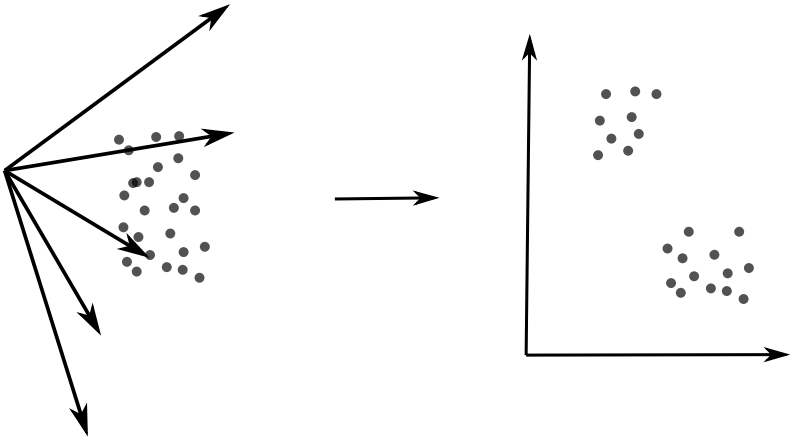
Artificial Neural Networks



Unsupervised learning – Applications

- Dimension reduction
- Clustering

Dimension reduction – basic idea



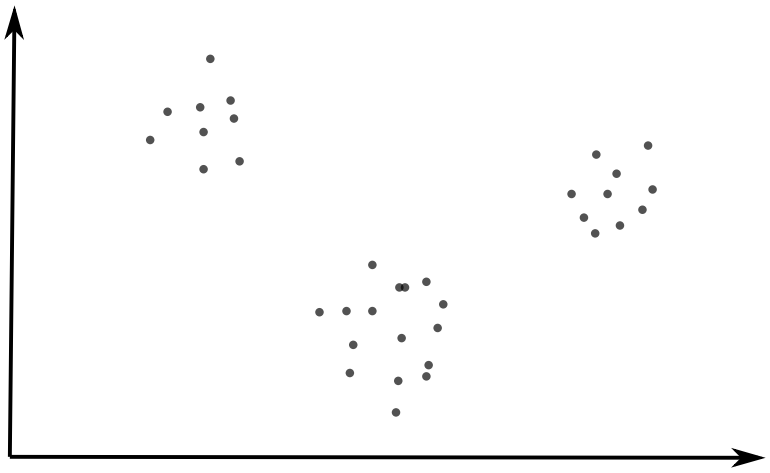
Dimension reduction – Applications

- Visualizations
- Feature selection

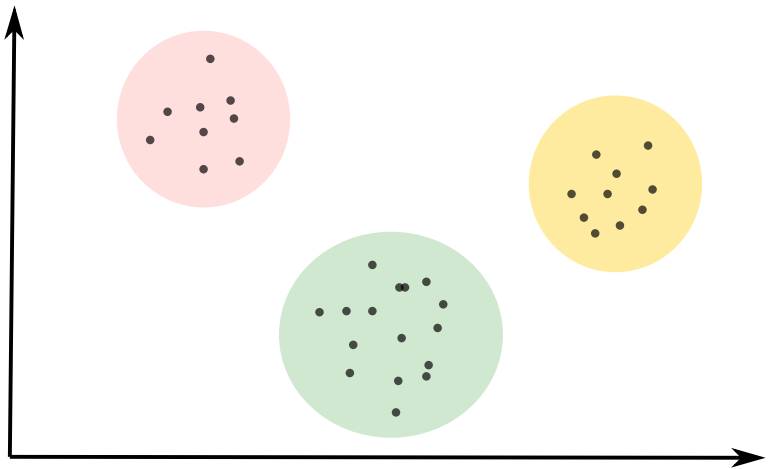
Dimension reduction – Selected methods

- PCA (Principle Component Analysis), lineaer
- t-SNE (t-distributed stochastic neighbor embedding), non-linear
- UMAP (Uniform manifold approximation and projection), non-linear

Cluster analysis



Cluster analysis



Cluster analysis – Selected methods

- k-means Clustering
- Hirachical Clustering
- DBSCAN (Density-based spatial clustering of applications with noise)

Thank you for your attention

konrad.foerstner.org / @kuf@mastodon.social / @konradfoerstner

zbmed.de / @ZB_MED

th-koeln.de / @th_koeln

ZB MED Information Centre for Life Sciences

Technology
Arts Sciences
TH Köln